Electronic Version
Stylesheet Version v1.1.1

# Claims

[c1] A method for creating circuit redundancy in programmable logic devices, the method comprising:identifying at least one single event upset sensitive sub-circuit of a programmable logic device;
introducing circuit redundancy for each single event upset sensitive sub-circuit identified.

[c2] The method of claim 1, wherein the step of identifying the at least one single event upset sensitive sub-circuit further comprises, identifying at least one single event upset sensitive gate.

[c3] The method of claim 2, wherein the step of identifying the at least one single event upset sensitive gate further comprises:
selecting a gate having a plurality of inputs;
selecting a threshold probability;
associating an input probability with each of the plurality of inputs;
assigning a logic value to each of the plurality of inputs, wherein a dominant logic value is assigned to the input if the input probability is greater than the threshold probability and a non-dominant logic value is assigned to the

input if the input probability is less than the threshold probability; and
determining the sensitivity of the gate.

[c4] The method of claim 3, wherein the step of determining the sensitivity of the gate further comprises, identifying the gate as a sensitive gate if only one of the plurality of inputs is assigned a dominant logic value.

[c5] The method of claim 3, wherein the step of determining the sensitivity of the gate further comprises, identifying the gate as a sensitive gate if all of the plurality of inputs are assigned non-dominant logic values.

[c6] The method of claim 3, wherein assigning a logic value to each of the plurality of inputs further comprises, assigning a logic value of zero to an input if the input probability is less than the threshold probability and assigning a logic value of one to an input if the input probability is greater than the threshold probability.

[c7] The method of claim 2, wherein the step of identifying at least one single event upset sensitive gate further comprises, identifying a gate as a sensitive gate wherein the gate is selected from the group consisting of EXOR, EXNOR and NOT gates.

[c8] The method of claim 1, wherein the step of introducing

circuit redundancy for each single event upset sensitive gate further comprises, introducing triple modular redundancy for each single event upset sensitive gate.

[c9] The method of claim 2, wherein the probability threshold is selected to satisfy a required single event upset immunity.

[c10] The method of claim 2, wherein the probability threshold is selected to satisfy a required area constraint of the programmable logic device.

[c11] The method of claim 2, wherein the step of identifying the at least one primary input probability further comprises software profiling.

[c12] The method of claim 1, wherein the step of identifying the at least one single event upset sensitive sub-circuit further comprises:
identifying a plurality of interconnected gates having a plurality of inputs and a plurality of outputs;
identifying a plurality of primary inputs;
identifying a plurality of primary outputs;
selecting a threshold probability;
associating an input probability with each of the plurality of primary inputs;
calculating an input probability for each of the plurality

of inputs of the plurality of interconnected gates by propagating the input probability of each of the plurality of primary inputs to the corresponding plurality of primary outputs;

assigning a logic value to each of the plurality of inputs and the plurality of primary inputs, wherein a dominant logic value is assigned to the input if the input probability is greater than the threshold probability and a non-dominant logic value is assigned to the input if the input probability is less than the threshold probability; and

identifying a single event upset sensitive sub-circuit by beginning at a primary output and backtracking recursively through the corresponding interconnected gates.

[c13] The method of claim 12, wherein the step of identifying a single event upset sensitive sub-circuit further comprises, identifying a gate in the sub-circuit as a sensitive gate if only one of the plurality of inputs to the gate is assigned a dominant logic value.

[c14] The method of claim 12, wherein the step identifying a single event upset sensitive sub-circuit further comprises, identifying a gate as a sensitive gate in the sub-circuit if all of the plurality of inputs to the gate are assigned non-dominant logic values.

[c15] The method of claim 12, wherein assigning a logic value

to each of the plurality of inputs and to each of the plurality of primary inputs further comprises, assigning a logic value of zero to an input if the input probability is less than the threshold probability and assigning a logic value of one to an input if the input probability is greater than the threshold probability.

[c16] The method of claim 9, wherein the step of identifying a single event upset sensitive sub-circuit further comprises, identifying a gate as a sensitive gate wherein the gate is selected from the group consisting of EXOR, EXNOR and NOT gates.

[c17] The method of claim 12, wherein the step of identifying a single event upset sensitive sub-circuit further comprises, identifying a gate as a sensitive gate when the output of the gate is a primary output of the interconnected gates.

[c18] The method of claim 12, wherein the step of identifying a single event upset sensitive sub-circuit further comprises, identifying at least one sensitive gate in the sub-circuit.

[c19] The method of claim 12, further comprising introducing triple modular redundancy for each gate of an identified single event upset sensitive sub-circuit.

[c20] The method of claim 19, further comprising implementing a voter between each triplicated gate and the input to a nontriplicated gate.

[c21] The method of claim 20, wherein the voter is implemented with a tri-state buffer.

[c22] The method of claim 12, wherein the probability threshold is selected to satisfy a required single event upset immunity.

[c23] The method of claim 12, wherein the probability threshold is selected to satisfy a required area constraint of the programmable logic device.

[c24] The method of claim 12, wherein the step of identifying the plurality of primary input probabilities further comprises software profiling.

[c25] A programmable logic device (PLD) configured to implement a circuit having reduced sensitivity to single event upsets, the circuit comprising a plurality of sub-circuits, the PLD comprising, a redundant circuit for each single event upset sensitive gate comprising each single event upset sensitive sub-circuit.

[c26] The PLD of claim 25, wherein the redundant circuit is a triple modular redundant circuit.

[c27] A method for reducing circuit sensitivity to single event upsets in programmable logic devices, the method comprising:
identifying a plurality of interconnected gates having a plurality of inputs and a plurality of outputs;
identifying a plurality of primary inputs;
identifying a plurality of primary outputs;
selecting a threshold probability;
associating an input probability with each of the plurality of primary inputs;
calculating an input probability for each of the plurality of inputs of the plurality of interconnected gates by propagating the input probability of each of the plurality of primary inputs to the corresponding plurality of primary outputs;
assigning a logic value to each of the plurality of inputs and the plurality of primary inputs, wherein a dominant logic value is assigned to the input if the input probability is greater than the threshold probability and a non-dominant logic value is assigned to the input if the input probability is less than the threshold probability;
identifying a single event upset sensitive sub-circuit by beginning at a primary output and backtracking recursively through the corresponding interconnected gates;
introducing triple modular redundancy for each gate of

an identified single event upset sensitive sub-circuit; and implementing a voter between each triplicated gate and the input to a nontriplicated gate, wherein the voter is implemented with a tri-state buffer.